Active Learning-based Online Coupling of Sawmill Simulators and Their Surrogate Model: The Effect of Sampling Bias on Concept Drift Detection

S Chabanet, Philippe Thomas, Hind Bril El Haouzi

(031)

Active Learning-based Online Coupling of Sawmill Simulators and Their Surrogate Model: The Effect of Sampling Bias on Concept Drift Detection

S. Chabanet, P. Thomas and H. Bril El-Haouzi
CRAN, Université de Lorraine, CNRS, Epinal, F-88000, France
E-mail: sylvain.chabanet@univ-lorraine.fr

Summary: Coupling numerical models with different computational costs and fidelity levels is a promising solution for developing efficient industrial digital twins able to process fast data streams. Such a coupling strategy between a high fidelity but computationally intensive simulation model and its machine learning-based surrogate model in the context of the sawmill industry had been proposed in a previous publication [1]. The strategy proposed is inspired by active learning and based on a measure of prediction confidence. It relies, however on the assumption that the input data stream models make predictions upon is stationary. Concept drifts are, however, frequent in an industrial context. It is, therefore, necessary to integrate into the proposed strategy a mechanism to detect such drifts. The present article evaluates the usage of several drift detection methods in conjunction with the previously proposed coupling strategy and highlights the issue caused by sampling bias. In particular, it can increase the time needed to detect the drift. Four datasets are used for evaluation. The first contains sawing simulation results, while the three others are benchmark datasets from the literature.

Keywords: Active learning, Sampling bias, Drift detection, Surrogate models.

1. Introduction

The coordination of digital models performing the same prediction task but varying in terms of fidelity level and computational requirement is an important step toward the integration of digital sobriety into operational digital twins [2]. While such twins might integrate high-fidelity simulation models, these tend, indeed, to be too computationally intensive to be used online over streams of data continuously collected from the physical twin. For example, [3] reports on the development of an industrial boiler digital twin integrating a multi-physics simulation model that requires five to seven days of computation to 1164 processors. A less extreme example is given by sawing simulators used in the forest-product industry which may need several minutes to simulate the sawing of a single log. This is, however, too slow for short-term decision problems that can involve several thousand such simulations. A common solution to lower the computational cost of such simulation models is to use surrogate models based on machine learning algorithms [4]. Such surrogate models, however, are only approximations of the original simulation model and cannot be expected to generalize as well, especially to scenarios never seen before. In the context of digital twins, it, therefore, appears necessary to retain the original simulation model to make predictions for a selected fraction of the data collected from the physical twin.

Such a strategy to couple a simulation model and its surrogate by actively selecting which model to use to predict labels for which data points collected from a physical twin has been proposed by [1]. This strategy is inspired by the theory of stream-based active learning (AL) [5] which studies how to actively sample data points from unlabeled data streams. In classic AL settings, these data points, and only these, are labeled and used to train ML models. The initial objective of such AL methods is to train ML models under restricted labeling budgets. Here, data points are selected from a data stream based on an estimation of a level of uncertainty in the surrogate prediction in order to minimize the prediction error of the coupled models. If this uncertainty level is too high, the simulation model is used instead of the ML model.

The strategy introduced by [1], however, relies on the assumption that the stream of data collected from the physical twin is stationary. This assumption ensures, in particular, that a specific simulation budget is targeted and that the performance of the surrogate model does not deteriorate. This assumption, however, might not be respected in industrial contexts, due, for example, to tool wear and changes in the characteristics of the raw material. Detecting concept drift in the input dataset stream would, therefore, be a first step toward the adaptation of the surrogate model.

This article discusses the integration of such a concept drift detection module in the strategy proposed by [1] and evaluates several drift detection strategies in this context. It additionally discusses and highlights a limitation of the proposed coupling strategy, which like many active learning-based methods causes sampling bias. This means that the statistical distribution of the data sampled to be labeled by the simulation model is not representative of the real distribution of the data [6]. It is known to cause various problems such as complicating the comparison and selection of ML models or undersampling important areas of the parameter space which leads in some cases

to poorly trained ML models. In the setting presented in this study, it can also, in some cases, impair drift detection.

The remainder of this article is organized as follows. Section 2 presents the context of this study and reviews previous works on coupling sawmill simulation models and their surrogate models. Section 3 then overviews the literature on concept drift detection and discusses the integration of these methods into the coupling strategy previously proposed by [1]. Section 4 presents experimental results evaluating several drift detection methods and highlights the problem caused by sampling bias. Section 5 concludes this paper.

2. Context

Many authors have studied the usage of sawing simulation models to support production planning and control in the forest-product industry [7]. These simulation models can be used to predict what set of lumbers would be obtained from sawing a specific log. Many authors, however, mention the important time taken by such simulations which limits their usage to mid and long-term production planning problems. In order to limit the computation time of these simulations and make possible their usage for short-term planning problems, [8], among others, proposed to use surrogate models of these simulators, that is, machine learning algorithms trained on past simulation results to perform the same prediction as the simulation model. Such surrogate models, however, remain only approximations of the original model, trained on relatively few data, and cannot be expected to generalize well to data inputs too different from the ones used to train them.

Recently, [1] proposed to couple both the simulation model and its surrogate model to benefit from their respective advantages as a step toward the development of a sawmill digital shadow able to support operational production planning. The objective is to decide for every data item transmitted by a stream whether the more precise simulation model should be used to replace the surrogate model prediction while targeting a user-defined simulation budget b.

The general workflow of this strategy is presented in Fig. 1. Consider a data stream transmitting new data items at random intervals. It is, here, modeled as a Poisson process. Similarly, the time required by the simulation model to predict the label of a data item is modeled as a random variable. It is assumed that any number of simulations can run in parallel. However, to limit computation costs, a budget corresponding to the average portion of data points sent to the simulator is targeted.

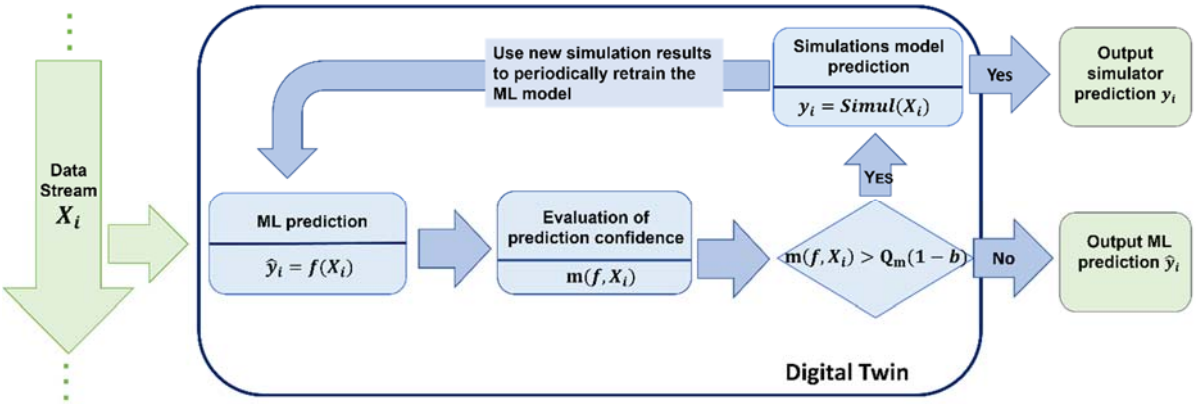

Fig. 1. Simplified framework of the coupling strategy proposed by [1].

For every data item X_i transmitted by the stream, a first prediction $\hat{y}_i$ is made by the machine learning model f. In this work, the machine learning models used are random forests, due to their good performances as sawmill simulators metamodels [7, 8]. A measure of uncertainty of the prediction is then computed. It has to be stressed out, however, at this stage the real label y_i of X_i is unavailable. The measure of uncertainty used here is, therefore, computed as the variance of the predictions of the trees in the forest:

$$\mathrm{m}(\mathrm{X}, \boldsymbol{f}) = \frac{1}{\mathrm{B}} \sum_{\mathrm{i}=1}^{\mathrm{B}} \left(f_i(X) - f(X) \right)^2$$

Here, $\boldsymbol{B}$ is the number of trees in the forest. The $\boldsymbol{f_j}$ are the infividual trees in the forest f. The quantile of this measure with respect to its overall distribution is then estimated from past values observed from the stream and compared with the threshold $1 - b$. This ensures that, as long as the stream remains stationary, the average portion of the stream that is sampled will tend toward b.

This strategy, however, relies on the assumption that the stream remains stationary to respect the simulation budget and ensure that past simulation results used to update the surrogate model remain representative of the prediction task.

3. Drift Detection

Many types of drifts have been identified in the related literature. In particular, they can be classified into real and virtual drift [11]. Real drift refers to the relationship between the inputs X_i and their labels y_i changing with time. Virtual concept drift refers to changes in the statistical distribution of the inputs X_i.

Drift detection methods often rely on monitoring univariate or multivariate streams updated with new data inputs. Different such streams can be used. Many authors proposed methods monitoring the prediction loss $l(y_i, \hat{y}_i)$. For example, [12] introduces a method based on control charts, often use in statistical process control, to monitor the error rate of neural networks. Similarly, the Page-Hinckley algorithm [13] is often used to monitor increases in these errors. Monitoring this error stream has the advantage that alerts can be raised only when concept drift results in a decrease in the model performance. It can also detect both real and virtual drift. However, it requires the obtention of the real targets y_i which, in practice, might not be obtained immediately, if at all. Other authors such as [14] propose to monitor the input feature stream, i.e., the stream of the X_i. More precisely, they propose a method based on the Kolmogorov-Smirnov statistical test between a sliding window of the stream and a fixed reference window. These data have the advantage of always being available. However, this stream is, more often than not, multivariate, which increases the risk of false alarms. Similarly, drift that does not impact the ML model performance can be detected and trigger useless updates of the model. In addition, purely real drift cannot be detected. Lastly, [15], for example, propose to use a stream composed of uncertainty measures of the predictions made by the ML model. More precisely, they use uncertainties associated with the predictions of Bayesian neural networks. Once again, this stream does not rely on the potentially unavailable targets. Additionally [15] argue that as long as the uncertainty measures are sufficiently correlated with the prediction loss $l(y_i, \hat{y}_i)$, detection will focus on drifts lowering the performance of the ML model. However, once again, it cannot detect purely real drifts.

These three types of streams can be monitored to detect drift in the previously described coupling strategy. The streams of the input features, X_i, and uncertainty measures, $\mathrm{m}(\mathrm{X}, \boldsymbol{f})$, are naturally part on the framework. The real targets y_i, however, are only available for the fraction of the inputs for which the simulation model is used, and only after a set amount of time corresponding to the time needed for the simulation to run.

While any of these streams could be monitored to detect concept drift, depending on the weight given to their respective advantages and disadvantages, the present article focuses on the stream of the prediction errors. While it is only partially available, it is the only type of stream allowing the detection of purely real drift and checking if the drift being detected effectively results in an increased prediction error.

4. Experiments

The performances of three drift detection procedures are evaluated to detect purely virtual drift from the partially measured stream of prediction losses. The first method is the control chart method (CC), introduced by [12]. The second is the method based on the Kolmogorov-Smirnov test used, for example, in [14]. This method depends on a parameter α corresponding to the risk of the test. The value of α used here was set to 0.01. Lastly, the Page-Hinckley method [13] is used. This method depends on two parameters, the so-called delta factor, and a detection threshold. Considering the sensitivity of the method to these parameters, whose scale might change depending on the dataset considered, they were all fixed here as multiple of the standard deviation σ of the errors observed on each dataset. More precisely, the delta factor was set, here, to 0.5σ for all experiments and the value of the threshold h was set to 10σ.

2.1. Datasets

Four datasets are used for the experiments presented in this paper.

The first dataset (SS) originates from the Canadian forest product industry. It contains information over 2219 real wood logs, including an estimate of the set of lumber sawed in it, simulated by the software Optitek. This set of lumber is modeled as a vector of integer of dimension 47 because up to 47 types of lumber can be sawed at the modeled sawmill. Every wood log is described by a vector of six know-how features commonly used in the industry. These features are the length, volume, curvature, diameters at both extremities, and shrinking of the log.

The three other datasets are from the UCI machine learning repository. They were selected because they cover regression tasks based on multivariate inputs and contain all more than 2000 points.

The first UCI dataset is the wine quality (WQ) dataset [16]. The task associated with this dataset is the prediction of wine qualities based on their physicochemical properties.

The second UCI dataset is the combined cycle power plant (CCPP) dataset [17]. Its aim is the prediction of a power plant production based on meteorological and physical data.

The last UCI dataset is the superconductivity data (SD) dataset [18] whose goal is to predict the critical temperature of superconductors based on their chemical formula.

None of these datasets naturally contains concept drift which allows controlling the type and time of drifts being artificially introduced during experiments.

Table 1. Detection rate of drift detection strategies for every dataset.

	SS		WQ		CCPP		SD	
	AL	RD	AL	RD	AL	RD	AL	RD
Kolmogorov-Smirnov	**1.0**	**1.0**	0.48	0.88	**0.99**	0.98	0.99	**1.0**
Page-Hinkley	0.87	0.91	0.54	0.84	**0.99**	0.95	0.97	0.95
Control-Chart	0.69	0.81	**1.0**	**1.0**	0.98	**0.99**	**1.0**	**1.0**

More precisely, the datasets were used to generate data drift composed of two stationary sections separated by an abrupt purely virtual drift. These streams are composed of a random ordering of the data points paired with random inter-arrival times sampled following an exponential law of rate $\lambda = 1$ to generate Poisson processes. In the second half of the stream, the targets of all data points were permutated at random to erase the relation between the input and output and create a purely real drift. Since the ML models will not be retrained during these experiments, there is no need to keep a structural relationship between the inputs and outputs past the drift.

2.1. Results

For every dataset and drift detection method, streams were generated following the previously described procedure and used as input to the coupling strategy described in Fig. 1. In addition, the partial error flow was used for drift detection. The time of the first alarm raised after the time of drift, if it exists, is considered as the time of detection. This was repeated 100 times for every dataset and drift detection method. To highlight the effect of the sampling bias on drift detection, this procedure was also repeated 100 more times but replacing the measure of prediction uncertainty used to select which samples to use the simulation model on by a random measure, so that the stream sampling process becomes independent of characteristics of the data points transmitted by the stream.

Table 2 presents the drift detection rate of every strategy on the four datasets. The drift detection rate corresponds, here, to the portion of the 100 streams that lead to a drift being detected. In this table, AL refers to the experiments using the measure of uncertainty $\mathrm{m}(.,\boldsymbol{f})$ to sample the stream, while RD refers to the experiments where this measure is replaced by randomly generated values. These rates are close to one, except for the Page-Hinkley and control chart detection methods on the SS dataset, and for the Kolmogorov-Smirnov and Pages-Hinckley methods on the WQ dataset. These two datasets are the shortest used for evaluation and might be too short for drift to be efficiently detected. Interestingly, however, these low detection rates are mostly observed for the AL stream sampling method which is a first indication that the sampling bias lower the performances of these drift detection methods on some datasets.

The median and interquartile range of the drift detection times for all drift detection methods and datasets are presented in Table 2. Medians are used due to the high standard deviations observed during experiments, making average estimates unstable. These experiments show larger detection times for the AL-based methods on the SS and WQ datasets. In particular, one-sided Brunner-Munzel comparison tests [19] were systematically run to evaluate the propension of the AL-based methods detection time at being higher than the baseline random method. These tests show significant differences at the 5 % level for all detection methods on the SS and WQ datatsets, except for the control-chart-based method on the WQ dataset. The test is also significant for the Page-Hinkley method on the SD dataset. This demonstrate that sampling bias can impair drift detection. It depends greatly, however, on the dataset and exact detection method used. The Page-Hinkley method, however, maintains rather low detection times even when affected by sampling bias.

Table 2. Medians and interquartile ranges of the drift detection time for every drift detection strategy and dataset.

	SS		WQ		CCPP		SD	
	AL	RD	AL	RD	AL	RD	AL	RD
Kolmogorov-Smirnov	241 (80)	224 (55)	963 (977)	587 (463)	209 (65)	206 (59)	289 (112)	270 (71)
Page-Hinkley	90 (108)	**49 (40)**	557 (758)	**180 (258)**	**31 (38)**	37 (42)	74 (48)	**59 (41)**
Control-Chart	111 (112)	66 (66)	402 (396)	313 (402)	331 (647)	289 (537)	151 (394)	325 (579)

5. Conclusion

This paper discusses the integration of a concept drift detection module in a previously proposed coupling strategy between a simulation model and its ML surrogate model. Several types of streams which can be monitored to detect drift are identified and their advantages and disadvantages are listed. Several drift detection methods are also tested on one of these drifts. Results point to the strong effect of the sampling bias

induced by the coupling strategy which tends to slow down significantly drift detection.

Table 3. Summarized characteristics of the four datasets used for evaluation.

Dataset name	Number of points	Number of features	Number of outputs
Sawmill Simulations (SS)	2219	6	47
Wine Quality (WQ)	4898	11	1
Combined Cycle Power Plant (CCPP)	9568	4	1
Superconductivity Data (SD)	21263	81	1

Future works will, therefore, have to find ways to mitigate this problem, either by investigating drift detection methods less sensitive to sampling bias or using statistical methods to correct the bias induced.

In addition, a single type of drift is studied here: abrupt real drift. The impact of sampling bias on other types of drift, and in particular of virtual drift, remain to be studied.

Acknowledgments

The authors gratefully acknowledge the financial support of the ANR-20-THIA-0010-01 Project LOR-AI (Lorraine intelligence artificielle) and région Grand EST.

We are also extremely grateful to FPInnovation who gathered and processed the dataset we are working with.

References

[1]. S. Chabanet, H. B. El Haouzi, Thomas, Toward a sawmill digital shadow based on coupled simulation and supervised learning models, in Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, *Springer*, Cham, 2023, pp. 59-70.

[2]. N. Julien, M. A. Hamzaoui, Integrating lean data and digital sobriety in digital twins through dynamic accuracy management, in Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, *Springer*, Cham, 2023, pp. 107-117.

[3]. J. P. Spinti, P. J. Smith, S. T. Smith, Atikokan Digital Twin: Machine learning in a biomass energy system, *Applied Energy*, Vol. 310, Mar. 2022, 118436.

[4]. S. Razavi, B. A. Tolson, D. H. Burn, Review of surrogate modeling in water resources, *Water Resources Research*, Vol. 48, Issue 7, 2012, W07401.

[5]. P. Kumar, A. Gupta, Active learning query strategies for classification, regression, and clustering: A survey, *J. Comput. Sci. Technol.*, Vol. 35, Issue 4, Jul. 2020, pp. 913-945.

[6]. R. Krishnan, A. Sinha, N. Ahuja, M. Subedar, O. Tickoo, R. Iyer, Mitigating sampling bias and improving robustness in active learning, *arXiv Preprint*, Sep. 13 2021, arXiv:2109.06321.

[7]. S. Chabanet, H. Bril El-Haouzi, M. Morin, J. Gaudreault, P. Thomas, Toward digital twins for sawmill production planning and control: Benefits, opportunities, and challenges, *International Journal of Production Research*, Vol. 61, Issue 7, Apr. 2023, pp. 2190-2213.

[8]. M. Morin, *et al.*, Machine learning-based models of sawmills for better wood allocation planning, *International Journal of Production Economics*, Vol. 222, Apr. 2020, 107508.

[9]. M. Morin, F. Paradis, A. Rolland, J. Wery, F. Laviolette, F. Laviolette, Machine learning-based metamodels for sawing simulation, in *Proceedings of the Winter Simulation Conference (WSC'15)*, Dec. 2015, pp. 2160-2171.

[10]. S. Wager, T. Hastie, B. Efron, Confidence intervals for random forests: The Jackknife and the infinitesimal jackknife, *J Mach Learn Res*, Vol. 15, Issue 1, Jan. 2014, pp. 1625-1651.

[11]. G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, F. Petitjean, Characterizing concept drift, *Data Min. Knowl. Disc.*, Vol. 30, Issue 4, Jul. 2016, pp. 964-994.

[12]. M. Noyel, P. Thomas, A. Thomas, P. Charpentier, Reconfiguration process for neuronal classification models: Application to a quality monitoring problem, *Computers in Industry*, Vol. 83, Dec. 2016, pp. 78-91.

[13]. D. V. Hinkley, Inference about the change-point from cumulative sum tests, *Biometrika*, Vol. 58, Issue 3, Dec. 1971, pp. 509-523.

[14]. R. Dos, M. Denis, P. Flach, S. Matwin, G. Batista, Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1545-1554.

[15]. L. Baier, T. Schlör, J. Schöffer, N. Kühl, Detecting concept drift with neural network model uncertainty, *arXiv Preprint*, Sep. 23 2022, arXiv:2107.01873.

[16]. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, *Decision Support Systems*, Vol. 47, Issue 4, Nov. 2009, pp. 547-553.

[17]. P. Tüfekci, Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods, *International Journal of Electrical Power & Energy Systems*, Vol. 60, Sep. 2014, pp. 126-140.

[18]. K. Hamidieh, A data-driven statistical model for predicting the critical temperature of a superconductor, *Computational Materials Science*, Vol. 154, Nov. 2018, pp. 346-354.

[19]. K. Neubert, E. Brunner, A studentized permutation test for the non-parametric Behrens-Fisher problem, *Computational Statistics & Data Analysis*, Vol. 51, Issue 10, Jun. 2007, pp. 5192-5204.